eIDAS-Node Migration Guide

For Version 2.4

Document history

Version	Date	Modification reason	Modified by
1.0	06/12/2019	Information how to migrate to eIDAS-Node v2.4 from eIDAS-Node v2.3.1	DIGIT

Disclaimer

This document is for informational purposes only and the Commission cannot be held responsible for any use which may be made of the information contained therein. References to legal acts or documentation of the European Union (EU) cannot be perceived as amending legislation in force or other EU documentation.

The document contains information of a technical nature and does not supplement or amend the terms and conditions of any procurement procedure; therefore, no compensation claim can be based on the contents of this document.

Table of contents

1. Introduction

This document is intended for a technical audience consisting of developers, administrators and those requiring detailed technical information on how to configure, build and deploy the eIDAS-Node application.

The purpose of this document is to facilitate migration from eIDAS-Node v2.3.1 to eIDAS-Node v2.4.

1.1. Document structure

This document is divided into the following sections:

Chapter 1 — *Introduction*: this section.

Chapter 2 — *Prerequisites*: Identifies any prerequisites that are required before migrating your eIDAS-Node to version 2.4.

Chapter 3 — *Changes*: Contains detailed information about the changes that should be taken into consideration when migrating to eIDAS-Node version 2.4.

1.2. Document aims

The main aim of this document is to provide information on all the changes requiring your action when migrating to eIDAS-Node version 2.4, including:

- configuration changes; and
- changes to code.

2. Prerequisites

Before starting your migration to eIDAS-Node version 2.4 you should have:

- already implemented eIDAS-Node version 2.3.1;
- downloaded the eIDAS-Node v2.4 Integration Package; and
- downloaded the latest documentation.

3. Changes

3.1. Summary of changes

- (EID-335) XML Encryption with SAML
- (EID-594) Key agreement method not supported for encryption of session keys
- (EID-657) Testing of CEF EidasNode
- (EID-674) Unable to decrypt (v2.2)
- (EID-803) BouncyCastle: reinstalling JCE providers is incompatible with the deployment of multiple applications on the same server sharing this JCE
- (EID-916) Redirect and Post location whitelists are not correctly validated
- (EID-919) SAMLResponse validation and decryption is broken
- (EID-922) SPCountry to LightRequest
- (EID-925) Error when receiving a name identifier format different than unspecified
- (EID-958) 201907-03 vulnerability
- (EID-959) 201907-04 vulnerability
- (EID-960) 201907-05 vulnerability
- (EID-975) Failure when NameIDPolicy of request and response don't match
- (EID-977) LightResponse generated at Generic Connector doesn't contain RelayState
- (EID-1001) Issuer URL in SAML messages can be instrumented, to some extent
- (EID-1013) Issue with decryption of SAML Response in Generic Connector
- (EID-1017) Support brainpool curves for SAML Metadata Signing

3.2. CVE-2019-18632

For more details, please check CVE-2019-18632

3.2.1. *Code changes*

The following files were changed:

EIDAS-Encryption/src/main/java/eu/eidas/auth/engine/xml/opensaml/CertificateUtil.java

EIDAS-SAMLEngine/src/main/java/eu/eidas/auth/engine/core/impl/AbstractProtocolSigner.java

Special cases for checking trust chain based on issuer in AbstractProtocolSigner.java method checkValidTrust were removed as well as unnecessary methods getIssuerCertificate and hasCredential.

It was also implemented the adding of all signature certificates to the to be trusted signature credential before trust chain validation using new method getAllSignatureCertificates on CertificateUtil to allow better validation of trust chain.

The method CertificateUtil#getIssuerX509Certificate was removed.

Related to this change new Junit test were added for checking certificate trust at CertificateUtilTest.java and also new test certificates replaced the previous ones.

TestEidasNodeMetadataTrustChain.java tests were adapted to new keystores so several new keystores were added at:

EIDAS-Node/src/test/resources/keystore/eIDASkeystores/

eidasKeyStore_Connector_CC_TrustedIntermediate.jks

eidasKeyStore_Connector_CC_TrustedIntermediateTrustedRoot.jks

eidasKeyStore_Connector_CC_TrustedLeaf.jks

eidasKeyStore_Connector_CC_TrustedRoot.jks

metadata.jks

metadata_TC_WITHOUT_INTERMEDIATE_CA_CERTIFICATE.jks

metadata_TC_WITHOUT_ROOT_CA.jks

metadataPlusExtraCertificate.jks

metadataSingleCertificate.jks

and previous xml files at EIDAS-Node/src/test/resources/keystore were updated

SignModule_METADATA_INTERMEDIATE_CA_ROOT_CA_TRUST.xml

SignModule_METADATA_INTERMEDIATE_CA_TRUST.xml

SignModule_METADATA_NODE_CERT_TRUST.xml

SignModule_METADATA_ROOT_CA_TRUST.xml

SignModule_METADATA_SINGLE_CERTIFICATE.xml

SignModule_METADATA_TC.xml

SignModule_METADATA_TC_WITHOUT_INTERMEDIATE_CA_CERTIFICATE.xml

SignModule_METADATA_TC_WITHOUT_ROOT_CA_CERTIFICATE.xml

SignModule_METADATA_TC_WRONG_ORDER_EXTRA_CERTIFICATE.xml

related to this tests were update accordingly.

3.2.2. *Configuration changes*

The following configuration files were added/changed/updated

EIDAS-Config/keystore

eidasKeyStore_Connector_CA.jks

eidasKeyStore_Connector_CB.jks

eidasKeyStore_Connector_CC.jks

eidasKeyStore_Connector_CD.jks

eidasKeyStore_Connector_CF.jks

eidasKeyStore_METADATA_TC.jks

eidasKeyStore_Service_CA.jks

eidasKeyStore_Service_CB.jks

eidasKeyStore_Service_CC.jks

eidasKeyStore_Service_CD.jks

eidasKeyStore_Service_CF.jks

EIDAS-Config/server

SignModule_Connector.xml

SignModule_Service.xml

3.3. 201907-03 vulnerability

In logMessage methods present in several classes, the incoming Http Servlet Request was used in the construction of a WebRequest to make the retrieval of the eIDAS request and response and light request response tokens be handled in a similar way as done in the rest of the code. This change was done in the following classes:

/EIDAS-Node/src/main/java/eu/eidas/auth/commons

IncomingRequest.java

/EIDAS-Node/src/main/java/eu/eidas/node/security

AbstractSecurityRequest.java

/EIDAS-Node/src/main/java/eu/eidas/node/connector/

SpecificConnectorRequestServlet.java

/EIDAS-Node/src/main/java/eu/eidas/node/service/

SpecificProxyServiceResponse.java

/EIDAS-Node/src/main/java/eu/eidas/node/logging/connector/messages/

ConnectorIncomingEidasResponseLogger.java

ConnectorIncomingLightRequestLogger.java

/EIDAS-Node/src/main/java/eu/eidas/node/logging/service/messages

ProxyServiceIncomingEidasRequestLogger.java

ProxyServiceIncomingLightResponseLogger.java

ProxyServiceOutgoingEidasResponseLogger.java

In relation with this change, the following Junit tests classes were also changed:

/EIDAS-Node/src/test/java/eu/eidas/node/logging/connector/messages

ConnectorIncomingEidasResponseLoggerTest.java

ConnectorIncomingLightRequestLoggerTest.java

/EIDAS-Node/src/test/java/eu/eidas/node/logging/service/messages

ProxyServiceIncomingEidasRequestLoggerTest.java

ProxyServiceIncomingLightResponseLoggerTest.java

ProxyServiceOutgoingEidasResponseLoggerTest.java

New Test methods were added to assert that the last token is processed in the incoming Logger related classes.

3.4. 201907-04 vulnerability

A fix was implemented to resolve the 201907-04 vulnerability.

3.4.1. *Code changes*

The SP_METADATA_URL("spmetadataurl"), was removed from eu.eidas.auth.commons.EidasParameterKeys. Also, in eu.eidas.node.auth.connector.AUCONNECTORSAML, the code which was using SP_METADATA_URL was removed.

3.5. Rework Message Logging implementation

3.5.1. *Code changes*

The following EIDAS-Node classes previously in eu.eidas.node.connector package are now in the EIDAS-Node package eu.eidas.node.logging.connector.

ConnectorIncomingEidasResponseLoggerFilter.java
ConnectorIncomingLightRequestLoggerFilter.java
ConnectorOutgoingEidasRequestLoggerFilter.java
ConnectorOutgoingLightResponseLoggerFilter.java

The following EIDAS-Node classes previously in eu.eidas.node.connector.messages package are now in the EIDAS-Node package eu.eidas.node.logging.connector.messages.

ConnectorIncomingEidasResponseLogger.java
ConnectorIncomingLightRequestLogger.java
ConnectorOutgoingEidasRequestLogger.java
ConnectorOutgoingLightResponseLogger.java

The following EIDAS-Node classes previously in eu.eidas.node.service package are now in the EIDAS-Node package eu.eidas.node.logging.service.

ProxyServiceIncomingEidasRequestLoggerFilter.java
ProxyServiceIncomingLightResponseLoggerFilter.java
ProxyServiceOutgoingEidasResponseLoggerFilter.java
ProxyServiceOutgoingLightRequestLoggerFilter.java

The following EIDAS-Node classes previously in eu.eidas.node.service.messages package are now in the EIDAS-Node package eu.eidas.node.logging.service.messages.

ProxyServiceIncomingEidasRequestLogger.java
ProxyServiceIncomingLightResponseLogger.java
ProxyServiceOutgoingEidasResponseLogger.java
ProxyServiceOutgoingLightRequestLogger.java

All the *Logger java classes have been refactored and are now inheriting from a new abstract class encapsulating common code for message logging, the AbstractLogger java class.

The *Logger class hierarchy is using the new enum MessageLoggerTag.java.

AbstractLogger and MessageLoggerTag are located at the root of the EIDAS-Node package eu.eidas.node.logging.

The existing MessageLoggerBean in the eu.eidas.node.logging and its corresponding unit tests have been removed.

For the MessageLoggerUtils class, the following was implemented:

1. Removal of the following class properties with their corresponding getters and setters:nodeProtocolEngineFactory,metadataFetcher, connectorControllerService, serviceControllerService and countryCode.
2. Removed properties in point 1 were also removed from the MessageLoggerUtils definition in the Spring Application Context.
3. Methods getProxyServiceSamlEngine, getConnectorSamlEngine and getEntityId were made private
4. Unused method retrieveProxyServiceAttributes has been removed.
5. MessageLoggerUtils class became final
6. All public methods in MessageLoggerUtils became final
7. Unit test MessageLoggerUtilsTest has been extensively reworked to ensure proper unit testing.

Modifications in Logger Filter activations:

The WebFilter for ProxyServiceOutgoingLightRequestLoggerFilter is now using the internal redirect

```
@WebFilter(filterName = "ConnectorOutgoingEidasRequestLoggerFilter",
        urlPatterns = {"/internal/colleagueRequestRedirect.jsp"}, dispatcherTypes =
{DispatcherType.FORWARD})
```

The WebFilter for ConnectorOutgoingEidasRequestLoggerFilter is now using the internal redirect

```
@WebFilter(filterName = "ConnectorOutgoingEidasRequestLoggerFilter",
        urlPatterns = {"/internal/colleagueRequestRedirect.jsp"}, dispatcherTypes =
{DispatcherType.FORWARD})
```

The WebFilter for ProxyServiceOutgoingEidasResponseLoggerFilter is now using the internal redirect

```
@WebFilter(filterName = "ProxyServiceOutgoingEidasResponseLoggerFilter",
        urlPatterns = {"/internal/connectorRedirect.jsp",
"/presentSamlResponseError.jsp",
                       "/InternalExceptionHandler"},
        dispatcherTypes = {DispatcherType.FORWARD, DispatcherType.ERROR}
)
```

3.6. Eid-803 BouncyCastle: reinstalling JCE providers is incompatible with the deployment of multiple applications on the same server sharing this JCE

3.6.1. *Code changes*

eu.eidas.auth.engine.core.impl.BouncyCastleBootstrap has been adapted to the configuration change and unit tests were added.

3.6.2. *Configuration changes*

When multiple applications dealing with Bouncycastle are deployed on the same application server, ClassCastException can be raised when redeploying one of the application. This is because a JCE is loaded at global JVM level and not only in the application class loader and therefore there is a discrepancy between the class loader of the JCE and its services. To disable the reinstallation of JCE Providers for these types of applications, the new system property -Dblock.security.provider.reinstall is introduced.

If block.security.provider.reinstall is missing or false, reinstallation of JCE providers can be triggered.

If block.security.provider.reinstall is true reinstallation of JCE providers is blocked.

3.7. Migrating from logback 1.1.2 to logback 1.2.0

3.7.1. *Code changes*

When logback released version 1.2.0, the Encoder interface was changed and is no longer expected to handle an OutputStream.

Due to the logback modification regarding Outputstream usage in Encoder, migrating to logback 1.2.0 revealed a synchronisation issue in eidas code when multiple threads were logging on the same file. This synchronization issue has been fixed.

The impacted file HashAndCounterGenerator is located in the eu.eidas.node.logging.integrity package.

Synchronization issue while logging will be automatically detected by the LoggingConcurrencyTest located in the corresponding eu.eidas.node.logging.integrity test package

3.7.2. *Configuration changes.*

Logback version updated in the eIDAS-Parent pom.xml

3.8. Inconsistent behaviour when replaying Incoming LightRequest and Incoming LightResponse

3.8.1. *Code changes*

For the LightRequest, an anti-replay cache was implemented in AUCONNECTORSAML. processSpRequest. The anti replay for the LightRequest is now in the AUCONNECTOR.getAuthenticationRequest and has been removed from the AUCONNECTORSAML class.

The anti replay for the LightResponse was not existing and is now implemented in SpecificProxyServiceResponse.handleExecute.

The method public Boolean checkNotPresentInCache(final String messageId) was added in AUNODEUtil to handle the anti-replay for LightResponse.

For the method, public Boolean checkNotPresentInCache(final String samlId, final String citizenCountryCode), the samId parameter has been renamed messageId.

The code changes were covered with unit test doPostWithAntiReplayTriggered in SpecificProxyServiceResponseTest and AUCONNECTORTestCase.

3.9. Key Agreement Support

Key Agreement support has been added with the version 1.0.5 of the library se.swedenconnect.opensaml:opensaml-security-ext

3.9.1. *Code changes*

For the encryption, we updated the SAMLAuthnResponseEncrypter class to use KeyEncryptionParameters that fits with the public key published in the Connector's metadata.

If the key type of the encryption certificate published in the Connector's metadata is an EC key, the SAMLAuthnResponseEncrypter will use Key Agreement otherwise it will use Key Transport.

We defined a new class (EidasDefaultSecurityConfiguration) to initialize the Opensaml context, this provides three methods:

- a preInitialize method to ensure the presence of the BouncyCastleProvider among the security providers (this can be blocked by setting the system propery "block.security.provider.reinstall" to true);
- an initialize method using the opensaml initialization service to initialize all;
- a postInitialize method to override opensaml security configuration with eidas security configuration defaults.

For the decryption, DecryptionParameters is used, which based on the credentials and the encrypted key from the response, will resolve the data key that will then use to decrypt the data of the EncryptedAssertion.

A DecryptionUtils class was created in order to build the DecryptionParameters.

The DecryptionParameters are defined with chaining resolvers, in order to handle multiple types of encrypted keys.

The AbstractProtocolDecrypter is also impacted since we can now profit from the fact that the decrypter can use an array of credentials, instead of determining ourself which private key to use, we load all private keys from the keystore and let the Decrypter determine based on the encrypted assertion which private key to use.

With the introduction of a new configuration property for the key encryption algorithm for key agreement done with a key wrapping algorithm, multiple classes have been modified, a new constructor with an additional parameter "keyEncryptionAlgorithmForKeyAgreement" has been added in the:

- AbstractProtocolEncrypter class
- EncryptionConfiguration class
- Static BaseProtocolEncrypter class from the AbstractSamlEngineEncryption class
- AbstractSamlEngineEncryption class

The constructors which don't provide a configuration for the keyEncryptionAlgorithmForKeyAgreement in those same classes are now deprecated.

The method getEncryptionConfiguration from the KeyStoreEncryptionConfigurator class has been modified to retrieve the new property "key.encryption.algorithm.key.wrapping" from the configuration file. The new property value is described by a new enum value (KEY_ENCRYPTION_ALGORITHM_FOR_KEY_AGREEMENT) of the EncryptionKey enum class.

In order to have a default encryption algorithm to be used when the node needs to encrypt with key agreement, the SAMLAuthnResponseEncrypter#Builder#validate method has been modified to set the value of the keyEncryptionAlgorithmForKeyAgreement instance variable to its default value based on the value of the new enum variable KEY_ENCRYPTION_ALGORITHM_FOR_KEY_AGREEMENT in the DefaultEncryptionAlgorithm enum class.

3.9.2. *Configuration changes*

In order to use Key Agreement, the properties "responseDecryptionIssuer" and "serialNumber" of the EncryptModule_Connector.xml file need to be updated to match with the EC Private Key Entry of the keystore defined by the property "keyStorePath".

We added a new private key in the Keystores holding private key entries. The new private key use the Elliptic Curve algorithm and is used for Key Agreement. This differs from the RSA algorithm used by the other private key present in the Keystores and which is used for Key Transport.

The keystores that have been updated are in the

/EIDAS-Config/keystore folder

the files concerned are the following ones:

eidasKeyStore_Connector_CA.jks (new key alias: speps-ka-ca-demo-certificate)

eidasKeyStore_Connector_CB.jks (new key alias: speps-ka-cb-demo-certificate)

eidasKeyStore_Connector_CC.jks (new key alias: speps-ka-cc-demo-certificate)

eidasKeyStore_Connector_CD.jks (new key alias: speps-ka-cd-demo-certificate)

eidasKeyStore_Connector_CF.jks (new key alias: speps-ka-cf-demo-certificate)

eidasKeyStore.jks (new key alias: local-ka-demo-certificate)

Also the corresponding configuration to use the new private key entries was also introduced as commented as an example of a working configuration in:

/EIDAS-Config/server/EncryptModule_Connector.xml

A new entry key "key.encryption.algorithm.key.wrapping" has been added to the configuration file in:
/EIDAS-Config/server/EncryptModule_Service.xml
The new entry defines the encryption algorithm that should be used in the case of encryptions with key agreement; it refers to the Key wrapping algorithm as defined in the eIDAS specification.

The entry "key.encryption.algorithm.key.wrapping" can be omitted in the configuration file, if so then the default value (http://www.w3.org/2001/04/xmlenc#kw-aes256) will be used as encryption algorithm for key agreement.

3.10. Upgrade BouncyCastle version from 1.60 to 1.64

3.10.1. *Code changes*

Changed the version of the BouncyCastle dependency in the EIDAS-Parent pom.xml from 1.60 to 1.64.

3.10.2. *Configuration changes*

Depending on the type of server used, this may require some server configuration adaptations.

On both Wildfly supported versions, 11 and 15, the configuration of the bouncy castle module needs to be updated. The jar file introduced in the server folder:

```
"/modules/system/layers/base/org/bouncycastle/main/"
```

folder need to be updated with the jar corresponding to the new 1.64 version of Bouncy Castle. The "module.xml" file present in the same folder should also be updated. Both files for both version can be found in subfolder /org/bouncycastle/main/ under /AdditionalFiles/Wildfly11 and /AdditionalFiles/Wildfly15 folders.

More details on the procedure to be applied, can be found in section 6. "Verifying the installation" of eIDAS-Node Installation and Configuration Guide.

3.11. Ignite default config with SSL

3.11.1. *Configuration changes*

A folder "ignite" was reintroduced in the EIDAS-Config folder with **demo** jks in a keystore folder.

Those jks are for test purposes only and should be replaced by other for production usage.

The configuration present in the igniteNode.xml and igniteSpecificCommunication.xml was updated to add the default SSL configuration by defining the "sslContextFactory" in the "igniteNode.cfg" bean definition.

If an IBM Java jre is used, the "sslContextFactory" bean configuration would need the following additional property:

<property name="keyAlgorithm" value="IBMX509" />

This was added commented out in both igniteNode.xml and igniteSpecificCommunication.xml files, to be commented out if necessary as mentioned.

3.12. Ignite default config with ExpiryPolicy

3.12.1. *Configuration changes*

The configuration for caches present in the igniteNode.xml and igniteSpecificCommunication.xml were updated with an expiryPolicy that will remove cache messages older than a certain period of time.
In igniteNode.xml cache duration vary in the following manner:

- anti replay caches: 3 hours;
- correlation cache: 7 minutes;
- metadata cache: 24 hours.

To configure the expiry policy add a property named "expiryPolicyFactory" inside each CacheConfiguration containing a bean of CreatedExpiryPolicy or using the reference to a prototype bean like so:

```
<bean class="org.apache.ignite.configuration.CacheConfiguration">
   …
   <property name="expiryPolicyFactory" ref="7_minutes_duration"/>
</bean>

<bean id="7_minutes_duration"
      class="javax.cache.expiry.CreatedExpiryPolicy" factory-method="factoryOf"
      scope="prototype">
    <constructor-arg>
        <bean class="javax.cache.expiry.Duration">
            <constructor-arg value="MINUTES"/>
            <constructor-arg value="7"/>
        </bean>
    </constructor-arg>
</bean>
```

Please check Appendix C of the eIDAS-Node Installation and Configuration Guide and Appendix D. Ignite proposed configuration from eIDAS-Node National IdP and SP Integration Guide for more details on the example configuration.

3.13. Update of hazelcast cache expiration duration to align with Ignite expiration values

To align with the changes in section 3.12, the same values were needed to be applied to Hazelcast configuration.

3.13.1. *Configuration changes*

The configuration for caches present in the hazelcastNode.xml and hazelcastSpecificCommunication.xml was updated with an expiryPolicy that will remove cache messages older than a certain period of time.
In hazelcastNode.xml cache duration vary in the following manner:

- anti replay caches: 3 hours;
- correlation cache: 7 minutes;
- metadata cache: 24 hours.

3.14. Fix for Redirect and Post location whitelists are not correctly validated

In version 2.3.1, the validation of an uri was being done by validating in both POST and REDIRECT uri whitelists. Therefore a fix was done to have only uri validation for the correct whitelist.

3.14.1. *Code changes*

In EidasNodeValidationUtil#validateConnectorDestination the method was modified specially in the way conditions for http method checking and uri validation is checked.

EidasNodeValidationUtilTest class was added, with several methods to the EidasNodeValidationUtil#validateConnectorDestination method.

In EIDASValues the following renaming was performed:

EIDAS_CONNECTOR_REDIRECT_URIDEST was renamed to EIDAS_CONNECTOR_REDIRECT_URI_DESTINATION_WHITELIST and

EIDAS_CONNECTOR_POST_URIDEST was renamed to EIDAS_CONNECTOR_POST_URI_DESTINATION_WHITELIST

3.14.2. *Configuration changes*

After this fix, it will not be necessary to set uri in both POST and REDIRECT whitelists for Connector destinations.

3.15. Fix whitelist control for Metadata Fetcher

The loggers of the Eidas-Node were not checking the metadata fetcher's whitelist before making the call to the metadata endpoint.

3.15.1. *Code changes*

The BaseMetadataFetcher class was modified to provide a control of the whitelist and two new beans were added in order to be used by the loggers with the whitelist properties specific to the node instance (connector or proxyservice).

3.16. Support for Http forwarding through Proxy

3.16.1. *Code changes*

The BaseMetadataFetcher class, which is the only place where the Node is making Http request, has been modified in order to take JVM properties into account.

3.16.2. *Configuration changes*

In order for the node to use a proxy, the following JVM properties can be used, depending on the needs:

- http.proxyHost
- http.proxyPort
- http.proxyUser
- http.proxyPassword
- http.nonProxyHosts

Note that the property http.nonProxyHosts could be needed to avoid using proxy for remote caches like ignite.

Example of proxy configuration without authentication:
`set "JAVA_OPTS=%JAVA_OPTS% -Dhttp.proxyHost=192.168.137.129 -Dhttp.proxyPort=8888 -Dhttp.nonProxyHosts=127.0.0.1"`

3.17. Conveying SPCountryCode value in LightRequest's Citizen Country Code field

In order to pass the Service Provider Country code in the Proxy-Service to Specific Proxy Service Light Request, a temporary implementation was done so that the Service Provider Country Code can be sent inside the Citizen Country Code of Light Request.

3.17.1. *Code changes*

The class ColleagueRequestServlet.java, was changed so that a new method updateCitizenCountryCodeValue is called from buildLightRequest to set the value of LightRequest.Builder.citizenCountry code by the one obtained from authenticationRequest. getOriginCountryCode().

A new test class ColleagueRequestServletTest.java, was added to test the behaviour for the current implementation.

EidasParameterKeys.java was changed, a new constant REPLACE_CITIZEN_COUNTRY_CODE_BY_SP_COUNTRY_CODE_IN_PROXYSERVICE_LIGHT_REQUEST was added.

In PropertiesUtil.java, a new method isReplaceCitizenCountryCodeBySpCountryCode was added to get the value from the external configuration entry which key is replace.citizenCountryCode.by.spCountryCode.in.proxyService.lightRequest.

3.17.2. *Configuration changes*

A new boolean value property replace.citizenCountryCode.by.spCountryCode.in.proxyService.lightRequest is available but not mandatory to be set. However, in eidas.xml the commented entry was added to exemplify how it could be set, if necessary.

<!--Boolean that replaces the citizen country code by sp country code on Proxy-Service's Light Request side if set to true-->

<!--<entry key="replace.citizenCountryCode.by.spCountryCode.in.proxyService.lightRequest">true</entry>-->

3.18. Add configuration property to enable/disable validation of prefixing identifier like attribute values

An external configuration property and implementation was added to enable/disable validation of prefixing identifier like attribute values.

3.18.1. *Code changes*

- AUCONNECTORSAML.java: added check for configuration to identifier pattern validation.

3.18.2. *Configuration changes*

Added entry in eidas.xml to enable or disable validation of prefixing eIDAS identifier-like attribute values, which key is:

- validate.prefix.country.code.identifiers

By default, this key's value will be set to true, this configuration needs only be used to explicitly disable the validation.

3.19. Add configuration property to enable/disable the prefixing with country identifiers

3.19.1. *Code changes*

- AUSERVICE.java: extracted methods for country code prefixing, encapsulated in check for the new configuration.

3.19.2. *Configuration changes*

Added entry in eidas.xml to enable or disable the prefixing with country codes eIDAS identifier-like attributes, which key is:

- insert.prefix.identifiers.country.code

By default this key will be set to true, this configuration needs only be used to explicitly disable the prefixing.

3.20. Failure when NameIDPolicy of Request and Response don't match

3.20.1. *Code changes*

The check of the match between NameIDPolicies of request and response has been restricted to the case where the NameIDPolicy of the request is not "unspecified".

The modification has been done in the AssertionUtil class

The sysadmin.properties file of the node has been changed to improve the invalid.idp.response message.

The error message is now "Invalid Light Response" instead of "Invalid SAML Response" which was misleading. This will avoid confusions concerning the place where the error has occurred.

3.21. Support brainpool curves for SAML Signing

3.21.1. *Code changes*

Added in the EidasDefaultSecurityConfiguration class, a new method setting the BouncyCastle as the provider to use for signatures, if the BouncyCastle provider is present.

This new method updateSignatureConfiguration is called during the post initialization of the EidasDefaultSecurityConfiguration.

3.21.2. *Configuration changes*

A keystore "eidasKeyStore_METADATA_EC.jks" has been added in the EIDAS-Config/keystore folder.

In addition to this newly added keystore, two new configuration files (SignModule_Service_EC.xml and SignModule_Connector_EC.xml) were added in the EIDAS-Config/server folder.

Those two new configuration files contain a configuration to sign with elliptic curves. The SamlEngine.xml file has also been modified to introduce the new SignModule configuration file as comments.

3.22. 201907-05 vulnerability

An external configuration property and implementation was added to enable/disable adding the value from “x-forwarded-for” header or remoteIpAddress to SubjectConfirmationData address attribute value in SAML response assertion.

Also, the possibility to get the value, to be put in already mentioned address attribute, from the "http-x-forwarded-for" header value was removed from the eIDAS code.

3.22.1. *Code changes*

-AssertionUtil.java: added check for configuration to add ipAddress to SubjectConfirmationData.

3.22.2. *Configuration changes*

Added value in eidas.xml to enable or disable adding ipAddress. The property is as configuration key as follows:

- enable.address.attribute.subject.confirmation.data

By default, this key will be set to false, setting it to true should only be used to explicitly add the value to address attribute value from either "x-forwarded-for" header or remoteIpAddress values.

3.23. Other fixes/improvements requiring no action

3.23.1. *Update eIDAS Metadata VERSION 2.4.0 Application Identifier*

Changed the eidas.application.identifier value from CEF:eIDAS-ref:2.3.1 to CEF:eIDAS-ref:2.4 in the external configuration file eidas.xml.

3.23.2. *Extracted method for duplicate code in processSpRequest from AUCONNECTORSAML*

Removed duplicated code, extracted to "validateServiceRedirectUrlValue" method in "AUCONNECTORSAML" class.

3.23.3. *Update eIDAS Metadata pom.xml*

Removed duplicated "org.slf4j" dependency.

3.23.4. *Update version for Maven Jar Plugin from eIDAS SAMLEngine pom.xml*

Added "${maven.jar.plugin.version}" version and "maven-jar-plugin" plugin in "pluginManagement" in EIDAS-Parent pom.xml.

3.23.5. *Remove Config profile from eIDAS Parent pom*

EIDAS-Config pom.xml was removed since the module had no code to compile, therefore the corresponding config was removed from the EIDAS-Parent/pom.xml

3.23.6. *Renamed jsp fields*

Renamed the id of the fields dummyField to redirectingMessageIdin the following jsp files:

EIDAS-Node/src/main/webapp/internal/colleagueRequestRedirect.jsp

EIDAS-Node/src/main/webapp/internal/connectorRedirect.jsp

EIDAS-Node/src/main/webapp/internal/tokenRedirectMsConnector.jsp

EIDAS-Node/src/main/webapp/internal/tokenRedirectMsProxyService.jsp

EIDAS-Node/src/main/webapp/internal/js/redirectOnload.js

3.23.7. *Update javaDoc*

It was necessary to update some Javadoc in order to resolve several warnings occurred when generating the Javadocs. Java 8 is more restricting on this topic than Java 7.

3.23.8. *Declaration and setting of local variable strSamlToken done in different subsequent lines*

`EIDAS-Node/src/main/java/eu/eidas/node/auth/connector/AUCONNECTORSAML`
.java

3.23.9. *Property active.module.connector should also disable incoming eIDAS SAML Response*

Blocking the eIDAS SAML Response if the parameter active.module.connector is set to false.

3.23.10. *Property active.module.service should also disable Proxy-Service response's entry point*

Blocking the SpecificProxyService response if the parameteractive.module.service is set to false.

3.23.11. *Fix self assignment in ConnectorControllerService# setConnectorRequestCorrelationCache*

There was a Self assignment issue in method ConnectorControllerService# setConnectorRequestCorrelationCache. This was fixed by changing the type of connectorRequestCorrelationCache from CorrelationMap<StoredAuthenticationRequest> to Cache<String, StoredAuthenticationRequest> and renaming parameter from method setConnectorRequestCorrelationCache from connectorRequestCorrelationMap to connectorRequestCorrelationCache. A test class ConnectorControllerServiceTestwas was added to cover ConnectorControllerService.

3.23.12. *Upgrade Shibboleth version from 7.3.0 to 7.5.0*

Shibboleth version updated in the eIDAS-Parent pom.xml

3.23.13. *Upgrade Opensaml version from 3.3.0 to 3.4.3*

Opensaml version and Opensaml api version are updated in the eIDAS-Parent pom.xml

3.23.14. *Add Opensaml extension se.swedenconnect.opensaml:opensaml-security-ext 1.0.5*

Opensaml security extension version 1.0.5 is added in the eIDAS-Parent pom.xml and dependency is added in the eidas-encryption module.

3.23.15. *Fix credential selection for Connector eIDAS SAML Response decryption*

Fix decryption problems linked to the presence of multiple privateKey entries in the Keystore. Instead of returning the first privateKey of the keystore, the correct private key depending on the key resolution of the EidasResponse will be used.

Code changes

- AbstractProtocolDecrypter.java: Retrieve all the private key entries from the keystore of the Connector and depending on the request (EidasReponse) the SAMLAuthnResponseDecrypter will use the adequate private key.
- CertificateUtil.java: Add the possibility to getCertificates of KeyInfo when KeyAgreement is used.

3.23.16. *Documentation update regarding responseToPointIssuer*

Fix of the description of the encryption property key “responseToPointIssuer” in eIDAS-Node and SAML document.

3.23.17. *Improving CSP behavior relying on security.header.CSP.report.uri*

If CSP is disabled the CSP header will not be included in the response header

If CSP is enabled and the uri specified for the CSP report-uri directive is valid, then CSP reporting is activated.

If CSP is enabled and the uri specified for the CSP report-uri directive is considered as being invalid, the report-uri directive will not be part of the CSP header included in the response.

For clarifications regarding a valid/invalid report-uri, please check the table "Security HTTP header parameters" in section "Additional consideration – Security" of the Node installation and configuration guide.

3.23.18. *Fix usage of session id in cookies for Weblogic*

The name of the cookie use to track the session in the weblogic was not correctly spelled which caused weblogic to append the sessionId to the end of the URL when encoding the urls.

The cookie-name has been fixed, by being set to "JSESSIONID".

3.23.19. *Reworked logging in test classes and ProcessLogin*

- Reverted error-level log of JAX exception back to debug-level log
- Removed catch blocks and added exception expect conditions where applicable in following test classes: AUCONNECTORUtilTestCase, EidasAuthRequestTest, EidasAuthResponseTest, TestEidasNodeFileMetadataProcessor, TestEidasNodeMetadataLoader

3.23.20. *Removed print of stacktrace from test*

- Print of stacktrace replaced by error-level logger of the or ExpectedException in several test classes:
- AUCONNECTORUtilTestCase.java
  EidasAuthRequestTest.java
  EidasAuthResponseTest.java
  ProcessLogin.java
  TestEidasNodeFileMetadataProcessor.java
  TestEidasNodeMetadataLoader.java

3.23.21. *Removed commented code*

- Commented code was removed from class SimpleProtocolProcess

3.23.22. *Preventing potential nullpointer exceptions in EIDAS-SAMLEngine*

- Encapsulated the configure() call of protocolProcessor in a null-check in method configureProtocolProcessor of class DOMConfigurator
- Extracted the protocolDecrypter to a separate variable for null-check in method getDecryptionCertificate in class AbstractProtocolEngine
- Improved the InstanceOf checks for the pub-priv metadata signing certificates by adding null-checking in class AbstractProtocolSigner

3.23.23. *Corrected typos in namespace prefixes in saml-engine-eidas-attributes.xml*

- eidas-reprentative-natural changed to eidas-natural-representative

- eidas-reprentative-legal changed to eidas-legal-representative

3.23.24. *Invalidate http sessions after successful request*

Since we are not using http sessions for something else than logging information about the request state when there is an issue, we have implement a process that will invalidate active sessions (if any) after the forwarding of request and response.

3.23.25. *Replace bouncy castle library*

For being able to build and deploy the EIDAS-Node in Weblogic 12C.1, the bouncy castle library bcprov-jdk15on version 1.64 has been replaced by the bcprov-jdk15to18 library version 1.64. A new profile weblogic12.1.3-BouncyCastle has been introduced in the EIDAS-Node pom, which is activated when a property named weblogic12.1.3-BouncyCastle is set in the maven build, if not the bcprov-jdk15on is used. For information on how to use this new profile, check section Weblogic server deployment in installation and configuration guide.

3.23.26. *Alllow only necessary settings of Consent attribute value from configuration files*

From SamlEgine_Connector.xml ,<entry key="consentAuthnResponse"> property was not necessary and it has been removed

From SamlEngine_Service.xml, <entry key="consentAuthnRequest"> property was not necessary and it has been removed

3.23.27. *Values from saml-engine-additional-attributes.xml are now trimmed*

The values from e.g. saml-engine-additional-attributes.xml are now trimmed avoid issues with e.g. CR\LF character endings.

3.23.28. *Setting of Incoming Connector's LightRequest RelayState in Connector's Outgoing LightResponse*

A fix was done in ColleagueResponseServlet#doPost, were a new method getRelayStateStoredEidasRequest is called, so that the Connector's outgoing LightResponse RelayState value is set with the RelayState value that was contained in the initial LightRequest received by the Connector and sent along side with the eIDAS SAML Request.

Added two Junit tests in relayStateInStoredEidasSamlRequestSetInLightResponse and relayStateInStoredEidasSamlRequestNullSetInLightResponse.

3.23.29. *Fix non deterministic build failure build of tests in EIDAS-SAMLEngine Module*

A non-deterministic build failure in the tests, in EIDAS-SAMLEngine module, was identified. The following classes: OpenSamlHelper, BouncyCastleBootstrapTest, EidasAuthRequestSignatureTest and EidasMessageFormatOnlyTest were updated to ensure BouncyCastle security provider is defined.

3.23.30. *Fix unrelevant logging error when decrypting with EC curves while having different kinds of credentials in keystore.*

Decrypting with KeyAgreement was sometimes logging an error, due to a SecurityException for failing to generate shared secret, even though it was decrypting successfully. Hence, this log was misleading.

In DecryptionUtils in the buildDefaultKeyInfoProviders, we replaced the KeyAgreementMethodKeyInfoProvider constructor by another constructor, which takes credentials as parameters and filters non EC ones out.

Several method signatures in DecryptionUtils were changed to allow passing the credentials to the new KeyAgreementMethodKeyInfoProvider constructor now used.

We also modified the buildDefaultKeyInfoCredentialResolver of the DecryptionUtils class to avoid reusing already configured KeyInfoProviders in the LocalKeyInfoCredentialResolver.